i.MX8 SHE API Rev 0.1

Generated by Doxygen 1.8.17

1 SHE API

This document is a software referece description of the API provided by the i.MX8 SHE solutions.

2 Revision History

Revision	date	description
0.1	Jul 06 2023	first draft

3 General concepts related to the API

3.1 Session

The API must be initialized by a potential requestor by opening a session.
The session establishes a route (MU, DomainID...) between the requester and the SHE module, and grants the usage of a specified key store. When a session is opened, the SHE module returns a handle identifying the session to the requester.

4 Module Index

4.1 Modules

Here is a list of all modules:

5 Module Documentation

5.1 Session

Data Structures

- struct she_hdl_s

Macros

- #define SHE_HANDLE_NONE (0x0)
- #define SHE_MAX_SESSIONS (16u)
- #define SHE_OPEN_SESSION_PRIORITY_LOW (0x00U)
- #define SHE_OPEN_SESSION_PRIORITY_HIGH (0x01U)
- #define SHE_OPEN_SESSION_FIPS_MODE_MASK BIT(0)
- #define SHE_OPEN_SESSION_EXCLUSIVE_MASK BIT(1)
- #define SHE_OPEN_SESSION_LOW_LATENCY_MASK BIT(3)
- #define SHE_OPEN_SESSION_NO_KEY_STORE_MASK BIT(4)

Typedefs

- typedef uint32_t she_hdl_t

Functions

- struct she_hdl_s * she_session_hdl_to_ptr (uint32_t hdl)
- void delete_she_session (struct she_hdl_s *s_ptr)
- struct she_hdl_s * add_she_session (void)

5.1.1 Detailed Description

5.1.2 Data Structure Documentation

5.1.2.1 struct she_hdl_s Structure describing the session handle members

Data Fields

struct plat_os_abs_hdl *	phdl	Pointer to OS device node.
uint32_t	session_handle	Session handle.
uint32_t	key_store_handle	handle to access key store
uint32_t	cipher_handle	handle to access cipher services
uint32_t	rng_handle	RNG handle.
uint32_t	utils_handle	handle to access utility
uint32_t	mu_type	Session MU type.

5.1.3 Macro Definition Documentation

5.1.3.1 SHE_HANDLE_NONE `#define SHE_HANDLE_NONE (0x0)`

Handle not available

5.1.3.2 SHE_MAX_SESSIONS `#define SHE_MAX_SESSIONS (16u)`

Maximum sessions supported

5.1.3.3 SHE_OPEN_SESSION_PRIORITY_LOW `#define SHE_OPEN_SESSION_PRIORITY_LOW (0x00U)`

Session opening priority flags Low priority. default setting on platforms that doesn't support sessions priorities.

5.1.3.4 SHE_OPEN_SESSION_PRIORITY_HIGH `#define SHE_OPEN_SESSION_PRIORITY_HIGH (0x01U)`

High Priority session.

5.1.3.5 SHE_OPEN_SESSION_FIPS_MODE_MASK `#define SHE_OPEN_SESSION_FIPS_MODE_MASK BIT(0)`

Operating Mode Only FIPS certified operations authorized in this session.

5.1.3.6 SHE_OPEN_SESSION_EXCLUSIVE_MASK `#define SHE_OPEN_SESSION_EXCLUSIVE_MASK BIT(1)`

No other SHE session will be authorized on the same security enclave.

5.1.3.7 SHE_OPEN_SESSION_LOW_LATENCY_MASK `#define SHE_OPEN_SESSION_LOW_LATENCY_MA↩ SK BIT(3)`

Use a low latency SHE implementation.

5.1.3.8 SHE_OPEN_SESSION_NO_KEY_STORE_MASK `#define SHE_OPEN_SESSION_NO_KEY_STORE_MA↩ SK BIT(4)`

No key store will be attached to this session. May provide better performances on some operation depending on the implementation. Usage of the session will be restricted to operations that doesn't involve secret keys (e.g. hash, signature verification, random generation)

5.1.4 Typedef Documentation

5.1.4.1 she_hdl_t `typedef uint32_t she_hdl_t`

Define the SHE handle type

5.1.5 Function Documentation

5.1.5.1 she_session_hdl_to_ptr() `struct she_hdl_s* she_session_hdl_to_ptr ( uint32_t hdl )`

Returns pointer to the session handle

Parameters

hdl	identifying the session handle.

Returns

pointer to the session handle.

5.1.5.2 delete_she_session() `void delete_she_session ( struct she_hdl_s * s_ptr )`

Delete the session

Parameters

s_ptr	pointer identifying the session.

5.1.5.3 add_she_session() `struct she_hdl_s* add_she_session ( void )`

Add the session

Returns

pointer to the session.

5.2 Key store

User must open a key store service flow in order to perform the following operations:

Macros

- #define MIN_MAC_LEN_NOT_SET BIT(0)
- #define MIN_MAC_LEN_SET BIT(1)
- #define KEY_STORE_OPEN_FLAGS_DEFAULT 0x0u
- #define KEY_STORE_OPEN_FLAGS_CREATE 0x1u
- #define KEY_STORE_OPEN_FLAGS_SHE 0x2u
- #define KEY_STORE_OPEN_FLAGS_SET_MAC_LEN 0x8u
- #define KEY_STORE_OPEN_FLAGS_STRICT_OPERATION 0x80u
- #define SHE_STORAGE_CREATE_SUCCESS 0u
- #define SHE_STORAGE_CREATE_WARNING 1u
- #define SHE_STORAGE_CREATE_UNAUTHORIZED 2u
- #define SHE_STORAGE_CREATE_FAIL 3u
- #define SHE_STORAGE_NUMBER_UPDATES_DEFAULT 300u
- #define SHE_STORAGE_MIN_MAC_BIT_LENGTH_DEFAULT 32u

Functions

- she_err_t she_open_key_store (she_hdl_t session_hdl, open_svc_key_store_args_t *args)

5.2.1 Detailed Description

User must open a key store service flow in order to perform the following operations:

- create a new key store
- perform operations involving keys stored in the key store (ciphering, signature generation...)
- perform a key store reprovisioning using a signed message. A key store re-provisioning results in erasing all the key stores handled by the SHE.

To grant access to the key store, the caller is authenticated against the domain ID (DID) and Messaging Unit used at the keystore creation, additionally an authentication nonce can be provided.

5.2.2 Macro Definition Documentation

5.2.2.1 MIN_MAC_LEN_NOT_SET `#define MIN_MAC_LEN_NOT_SET BIT(0)`

Minimum mac length not set

5.2.2.2 MIN_MAC_LEN_SET `#define MIN_MAC_LEN_SET BIT(1)`

Minimum mac length is set

5.2.2.3 KEY_STORE_OPEN_FLAGS_DEFAULT `#define KEY_STORE_OPEN_FLAGS_DEFAULT 0x0u`

default flags

5.2.2.4 KEY_STORE_OPEN_FLAGS_CREATE `#define KEY_STORE_OPEN_FLAGS_CREATE 0x1u`

Create a key store

5.2.2.5 KEY_STORE_OPEN_FLAGS_SHE `#define KEY_STORE_OPEN_FLAGS_SHE 0x2u`

Target key store is a SHE key store

5.2.2.6 KEY_STORE_OPEN_FLAGS_SET_MAC_LEN `#define KEY_STORE_OPEN_FLAGS_SET_MAC_LEN 0x8u`

Check min mac length

5.2.2.7 KEY_STORE_OPEN_FLAGS_STRICT_OPERATION `#define KEY_STORE_OPEN_FLAGS_STRICT_OP↩ ERATION 0x80u`

The request is completed only when the key store has been written in the NVM and the monotonic counter has been updated. This flag is applicable for CREATE operation only

5.2.2.8 SHE_STORAGE_CREATE_SUCCESS `#define SHE_STORAGE_CREATE_SUCCESS 0u`

New storage created successfully.

5.2.2.9 SHE_STORAGE_CREATE_WARNING `#define SHE_STORAGE_CREATE_WARNING 1u`

New storage created but its usage is restricted to a limited security state of the chip.

5.2.2.10 SHE_STORAGE_CREATE_UNAUTHORIZED `#define SHE_STORAGE_CREATE_UNAUTHORIZED 2u`

Creation of the storage is not authorized.

5.2.2.11 SHE_STORAGE_CREATE_FAIL `#define SHE_STORAGE_CREATE_FAIL 3u`

Creation of the storage failed for any other reason.

5.2.2.12 SHE_STORAGE_NUMBER_UPDATES_DEFAULT `#define SHE_STORAGE_NUMBER_UPDATES_DEFAU↩ LT 300u`

default number of maximum number of updated for SHE storage.

5.2.2.13 SHE_STORAGE_MIN_MAC_BIT_LENGTH_DEFAULT `#define SHE_STORAGE_MIN_MAC_BIT_LENG↩ TH_DEFAULT 32u`

default MAC verification length in bits

5.2.3 Function Documentation

5.2.3.1 she_open_key_store() `she_err_t she_open_key_store (`

```
she_hdl_t session_hdl,
open_svc_key_store_args_t * args )
```

Open a service flow on the specified key store.

Parameters

session_hdl	pointer to the handle identifying the current session.
args	pointer to the structure containing the function arguments.

Returns

error code.

5.3 Utils

User must open a SHE utils service flow in order to perform the following operations:

Data Structures

- struct op_open_utils_args_t

Functions

- she_err_t she_open_utils (she_hdl_t session_hdl, op_open_utils_args_t *args)

5.3.1 Detailed Description

User must open a SHE utils service flow in order to perform the following operations:

- Create a utils handle
- perform SHE key update extension
- update SHE plain key
- export SHE plain key
- get SHE identity (UID)
- get SHE status register
- perform MAC generation and verification in fast mode for a SHE session on V2X
- perform MAC generation and verification in fast mode for a SHE session

5.3.2 Data Structure Documentation

5.3.2.1 struct op_open_utils_args_t Structure describing the open utils service operation arguments

Data Fields

uint32_t	utils_handle	

5.3.3 Function Documentation

5.3.3.1 she_open_utils()

```
she_err_t she_open_utils (
        she_hdl_t session_hdl,
        op_open_utils_args_t * args )
```

Open SHE utils service flow on the specified key store. The SHE utils service flow can be opened only after opening SHE key storage handle.

Parameters

session_hdl	pointer to the handle identifying the current session.
args	pointer to the structure containing the function arguments.

Returns

error code.

5.4 Shared Buffer

Data Structures

- struct op_shared_buf_args_t

5.4.1 Detailed Description

5.4.2 Data Structure Documentation

5.4.2.1 struct op_shared_buf_args_t Structure describing the get shared buffer operation arguments

Data Fields

uint16_t	shared_buf_offset	offset of the shared buffer in secure memory
uint16_t	shared_buf_size	size in bytes of the allocated shared buffer

5.5 Error codes

Error codes returned by SHE functions.

Enumerations

- enum she_err_t {
  SHE_NO_ERROR = 0x0,
  SHE_SEQUENCE_ERROR = 0x1,
  SHE_KEY_NOT_AVAILABLE = 0x2,
  SHE_KEY_INVALID = 0x3,
  SHE_KEY_EMPTY = 0x4,
  SHE_NO_SECURE_BOOT = 0x5,
  SHE_KEY_WRITE_PROTECTED = 0x6,
  SHE_KEY_UPDATE_ERROR = 0x7,
  SHE_RNG_SEED = 0x8,
  SHE_NO_DEBUGGING = 0x9,
  SHE_BUSY = 0xA,
  SHE_MEMORY_FAILURE = 0xB,
  SHE_GENERAL_ERROR = 0xC,
  SHE_UNKNOWN_WARNING = 0x27,
  SHE_FATAL_FAILURE = 0x29 }

5.5.1 Detailed Description

Error codes returned by SHE functions.

5.5.2 Enumeration Type Documentation

5.5.2.1 she_err_t `enum she_err_t`

Error codes returned by SHE functions.

Enumerator

SHE_NO_ERROR	Success.
SHE_SEQUENCE_ERROR	Invalid sequence of commands.
SHE_KEY_NOT_AVAILABLE	Key is locked.
SHE_KEY_INVALID	Key not allowed for the given operation.
SHE_KEY_EMPTY	Key has not beed initialized yet.
SHE_NO_SECURE_BOOT	Conditions for secure boot process are not met.
SHE_KEY_WRITE_PROTECTED	Memory slot for key has been write-protected.
SHE_KEY_UPDATE_ERROR	Key update failed due to errors in verification of the messages.
SHE_RNG_SEED	The seed has not been initialized.
SHE_NO_DEBUGGING	Internal debugging is not possible.
SHE_BUSY	A function of SHE is called while another function is still processing.
SHE_MEMORY_FAILURE	Memory error (e.g. flipped bits).
SHE_GENERAL_ERROR	Error not covered by other codes occurred.
SHE_UNKNOWN_WARNING	SHE Unknown Warning.
SHE_FATAL_FAILURE	A fatal failure occurred, SHE goes in unrecoverable error state not replying to further requests

Index